



# Guided Creation and Update of Objects in RDF(S) Bases

Alice Hermann, Sébastien Ferré, Mireille Ducassé

## ► To cite this version:

Alice Hermann, Sébastien Ferré, Mireille Ducassé. Guided Creation and Update of Objects in RDF(S) Bases. International Conference on Knowledge Capture, Jun 2011, Banff, Canada. hal-00654084

**HAL Id: hal-00654084**

**<https://inria.hal.science/hal-00654084>**

Submitted on 20 Dec 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Guided Creation and Update of Objects in RDF(S) Bases

Alice Hermann  
IRISA – INSA Rennes  
Campus de Beaulieu  
35042 Rennes cedex  
France  
alice.hermann@irisa.fr

Sébastien Ferré  
IRISA – Université Rennes 1  
Campus de Beaulieu  
35042 Rennes cedex  
France  
sebastien.ferre@irisa.fr

Mireille Ducassé  
IRISA – INSA Rennes  
Campus de Beaulieu  
35042 Rennes cedex  
France  
mireille.ducasse@irisa.fr

## ABSTRACT

Updating existing knowledge bases is crucial to take into account the information that are regularly discovered. However, this is quite tedious and in practice Semantic Web data are rarely updated by users. This paper presents UTILIS, an approach to help users create and update objects in RDF(S) bases. While creating a new object,  $o$ , UTILIS searches for similar objects, found by applying relaxation rules to the description of  $o$ , taken as a query. The resulting objects and their properties serve as suggestions to expand the description of  $o$ .

## Categories and Subject Descriptors

I.2.4 [ARTIFICIAL INTELLIGENCE]: Knowledge Representation Formalisms and Methods

## General Terms

Human factors

## Keywords

Semantic Web, RDF(S) data, interactive knowledge acquisition, semantic data management

## 1. INTRODUCTION

Updating existing Semantic Web (SW) data is crucial to take into account information that are regularly discovered. This is, however, quite tedious and, in practice, SW data are rarely updated by users. In the Web 2.0, users are totally integrated in data production, with tags for example. Although motivation is not a problem, to directly enter data in SW formats is difficult. There exist tools to help users create and update knowledge bases [1, 3]. They lead to the creation of triples that are syntactically correct and consistent with the rules and axioms of the base. The knowledge used to assist users can be of several types (syntax, vocabulary, axioms of the ontology) and depends on the editor. However, when creating a new object, users benefit neither from existing objects and their properties, nor from the already known properties of the new object. For example, in a genealogical base, assume that a user creates a new person for whom the mother has already been entered. That fact can be used to guide the user in the choice of the father.

Transformation rules	
$r_1 \ p \ o$	$?x = r_1. \ ?x \ p \ o$
$s \ p \ r_2$	$s \ p \ ?x. \ ?x = r_2.$
$r_1 \ \text{rdf:type} \ c$	$?x = r_1. \ ?x \ \text{rdf:type} \ c.$

Figure 1: Transformation rules to get elementary triples. Variables have a leading question mark,  $r_i$  are resources,  $p$  is a property,  $c$  is a class;  $o$  and  $s$  are resources or variables.

Indeed, if the mother has another child, it is quite possible that that child and the new person have the same father.

This paper presents a method that uses the existing objects and the current partial description to help users create and update objects : UTILIS (Updating Through Interaction in Logical Information Systems).

## 2. RELAXATION OF THE DESCRIPTION

The description of an object is a conjunction of RDF(S) triples. UTILIS uses the current description of the new object as a query, to find similar objects. UTILIS suggests the objects similar to the new object and their properties as completions to the current description of the new object. For example, for the new person *MayaD*, the current description, in the Turtle concise syntax, is : *<MayaD> a :woman*. This description is taken as a query. For reasons of space, we abbreviate ‘SELECT ?x WHERE ?x’ by ‘what’ in SPARQL queries. The query corresponding to this description is *what a woman*. The results, all the women and their properties (e.g. *:motherOf []*, *:firstname []*), are used as suggestions.

The description of the new object quickly becomes unique in the knowledge base. For example, once the user has specified the firstname and the mother of *MayaD*, it is unlikely that another person has all those properties. In that case it does not match any other object, and the result of the query is, thus, empty. We propose to generalize the query derived from the description and to use the results of those queries as suggestions. To compute the generalized queries, we have defined relaxation rules, inspired by the query approximation rules of Hurtado et al. [4]. Figure 1 shows the transformation rules to split the query into elementary elements. The relaxation rules are then applied on those elementary elements. Figure 2 shows the current set of relaxation rules. The first column shows the name of the rule, the second column shows the triple before re-

Rule	Initial triple	Condition	Relaxed triple
Resource	$?r_1 = r_2$		nil
Super-Property	$?r_1 p_1 ?r_2$	$p_1 \text{ sp } p_2$	$?r_1 p_2 ?r_2$
Property	$?r_1 p_1 ?r_2$	$\#p \neq p_1.(p_1 \text{ sp } p)$	nil
Super-Class	$?r_1 \text{ type } c_1$	$c_1 \text{ sc } c_2$	$?r_1 \text{ type } c_2$
Class	$?r_1 \text{ type } c_1$	$\#c \neq c_1.(c_1 \text{ sc } c)$	nil

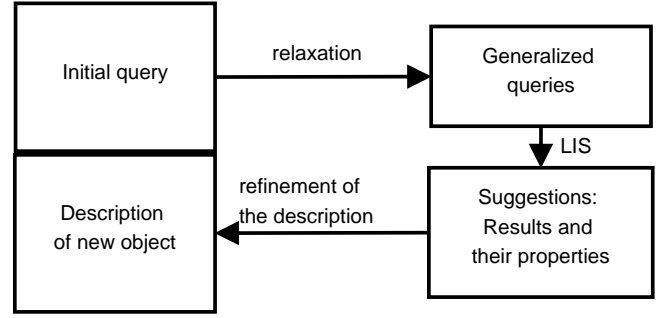
**Figure 2: Elementary relaxation rules.** Variables have a leading question mark,  $r_i$  are resources,  $p_j$  are properties, and  $c_k$  are classes; a relaxed triple set to nil means that the initial triple is actually removed. type, sc, sp are abbreviations for `rdf:type`, `rdfs:subClassOf` and `rdfs:subPropertyOf`

laxation, the third column indicates the conditions under which the rule can be applied, and the fourth column shows the triple after relaxation. For example, assume that MayaD has the following description, `<MayaD> a :woman; :first-name 'Maya'; :mother <TaraD>`. The user searches here to extend the description of MayaD. The query corresponding to this description is `what a :woman; :firstname 'Maya'; :mother <TaraD>`. That query has no answer. A possible generalized query, by application of the SuperClass rule on the first triple and the Resource rule on the third triple, is: `what a :person; :firstname []; :mother <TaraD>`. The result of that relaxed query is the sibling of MayaD and his properties (e.g. `:father [], :birthDate []`). After selecting `:father []`, the description is `<MayaD> a :woman; :firstname 'Maya'; :mother <TaraD>; :father []`. The user now searches for the father of MayaD. A possible generalized query is: `what is :father of [a :person; :first-name []; :mother <TaraD>]`. The result is the father of the sibling of MayaD, and thus the father of MayaD. After selecting this suggestion, the description is `<MayaD> a :woman; :firstname 'Maya'; :mother <TaraD>; :father <AlexD>`.

### 3. AN INCREMENTAL GUIDANCE

UTILIS reuses the interaction mechanisms of Camelis 2<sup>1</sup>, an implementation of logical information systems for the Semantic Web [2], for the incremental and interactive creation and update of objects. Camelis 2 supports exploratory search, guiding users from query to query, following navigation links. At each navigation step, navigation links are dynamically computed from the query. The key difference of UTILIS w.r.t. exploratory search is that navigation links are computed from a set of generalized queries derived from the partial description of the new object. Every generalized query has an associated distance that corresponds to the number of relaxations made. The results of generalized queries, similar objects, are sorted by distance. By default, only the results corresponding to the smallest distance are listed, but that list can be extended to greater distances upon user's request. The list of results may be extended until all objects are suggested, thus ensuring the completeness of the suggestions w.r.t. the knowledge base.

<sup>1</sup><http://www.irisa.fr/LIS/ferre/camelis/camelis2.html>



**Figure 3: Interactive refinement of the description of a new object.**

Figure 3 shows the interactive process to refine the description of a new object. The initial element of the process is the partial description of the new object. This description is used as the initial query to generalize the set of similar objects as needed. The resulting objects of the initial and generalized queries, as well as their properties, are used as suggestions. Those suggestions allow to extend the description of the new object. That new description is used as a new initial query and so on.

### 4. CONCLUSION

We propose a system to interactively and incrementally guide users in creating and updating objects in a RDF(S) graph. UTILIS is incremental, the guidance builds on the existing objects, and on the current description of the object being created or updated. Whenever an element is added to the description, it is used to compute suggestions for additional elements. UTILIS is also interactive, the user has an active part in the process. The guidance builds on the existing objects, and on the current description of the object being created or updated. Whenever an element is added to the description, it is used to compute suggestions for additional elements.

Users no longer need to enter the whole information. They, thus, do not need to know all the existing resources. They are less likely to forget to fill important properties. Using the existing objects and the current description of the object being edited, the guidance adapts to each object. In addition to comfort, our method ensures that the vocabulary used throughout the database is consistent.

### 5. REFERENCES

- [1] A. Bernstein and E. Kaufmann. Gino - a guided input natural language ontology editor. In *Proceedings of the International Semantic Web Conference (ISWC)*. Springer, November 2006.
- [2] S. Ferré. Conceptual navigation in RDF graphs with SPARQL-like queries. In *International Conference on Formal Concept Analysis (ICFCA)*, volume 5986 of *Lecture Notes in Computer Science*, pages 193–208. Springer, 2010.
- [3] H. Haller. QuiKey – an efficient semantic command line. In *Knowledge Engineering and Management by the Masses (EKAW)*, pages 473–482. Springer, 2010.
- [4] C. A. Hurtado, A. Poulouvasilis, and P. T. Wood. Query relaxation in RDF. *J. Data Semantics*, 10:31–61, 2008.